Power and Energy Normalized Speedup Models for Heterogeneous Many Core Computing

Mohammed A. N. Al-hayanni[1], Ashur Rafiev[2], Rishad Shafik[1], Fei Xia[1]
School of EEE[1] and CS[2], Newcastle University
Newcastle Upon Tyne, NE1 7RU, UK
E-mail: m.a.n.al-hayanni, ashur.rafiev, rishad.shafik, fei.xia@ncl.ac.uk

***Abstract*—Continued technology scaling in VLSI has enabled more and more computation cores to be integrated in the same chip. This has facilitated the parallelization of processing and the increase of performance whilst keeping energy consumption at reasonable levels. To study the potential improvement of performance in such many core systems, three existing models have been popular in both the research community and industry. Amdahl's law is the original speedup model that estimates the maximum performance improvement with fixed workloads. Gustafson's law is a popular model that introduces variable workloads and estimates fixed time speedup. Sun and Ni combined the above two models into one considering the memory-bounded situation. These models are further extended via the Hill-Marty model to cover a limited form of heterogeneity. This paper extends these models to cover a more comprehensive assumption of core heterogeneity. We also present power and energy models based on the extended heterogeneous models. Our models cover popular power and performance control methods such as Dynamic Voltage Frequency Scaling (DVFS), power gating, etc. A case study is performed with an ARM big.LITTLE architecture containing Cortex A7 and A15 cores, including a comprehensive analysis with different ratios of parallel and sequential workloads to identify the most energy-efficient system configuration based on these models. Experimental results demonstrated high correlations between practically measured power normalized performance and that of the proposed extended models.**

***Index Terms*—many-core processors; speedup; energy-efficiency; power and energy normalized performance.**

I. Introduction

Technology scaling has facilitated significant performance improvement at reduced power consumption through increased operating frequency and smaller device geometries [1]. According to Dennard's CMOS scaling law [2] despite such smaller geometries the power density of these devices remains constant. This is because the number of transistors per unit of area is also increasing substantially, which also conforms to Moore's [3] and Koomey's laws [4]. Dennard's law further states that the performance per watt is growing exponentially, doubling every 1.5 years.

Over the years significant research has been carried out to understand the trend of performance growth with many interconnected cores. An examples of these models is Pollack's Rule, which suggests that performance is increasing approximately proportional to the square root of the complexity [5]. Following this rule, doubling the number of processors also doubles the performance [1]. Therefore, multi-core systems will deliver further improvement in throughput and latency for the same die area.

The most appropriate metric to describe performance gain is speedup. The first scalable model in relation to the multi-core processor model is explained by Amdahl's law [6]. It assumes that a fixed workload is executed in N processors of a multi- or many-core system and compares the throughput/performance with the same workload executed in a single processor. In 1988, Gustafson introduced the principle of scalable computing in multi-core processors pertaining to the fixed time model. This model proposes a linear speedup model that increases the workload proportional to increasing machine scalability, while the execution time remains fixed [7]. In other words, more parallel processors complete larger workloads spending the same amount of time and the speedup is calculated according to how much larger the workload is in multiple cores compared with that in a single core. In 1990, Sun and Ni suggested a new model, which included extended workload calculations by considering the capability of the memory. It is important to note that the executed workload and time should change based on the capability of the system, while the performance calculations appeared linear within the increasing cores [8], [9].

TABLE I: Existing Speedup Models and the Proposed Model

	Homogeneity	Heterogeneity	Power	Amdahl's Model	Gustafson's Model	Sun and Ni's Model
[6]	Yes	No	No	Yes	No	No
[7]	Yes	No	No	Yes	Yes	No
[8]	Yes	No	No	Yes	Yes	Yes
[14]	Yes	Simple	No	Yes	No	No
[15]	Yes	Simple	No	Yes	Yes	Yes
[16]	Yes	Simple	Yes	Yes	No	No
[17]	Yes	No	No	Yes	Yes	Yes
Extended Model	Yes	Normal form (Section III)	Yes	Yes	Yes	Yes

On the other hand, power consumption management is a significant issue in scalable systems. For instance, DVFS, clock gating and power gating techniques are designed for this reason. The fine grain power management suggested by [1], [10] [11] [12] [13] are some of the scaling techniques used in order to decrease power consumption.

Speedup models described in existing studies for the comprehensive understanding of core modeling are listed in Table I. The Hill-Marty model extended Amdahl's law to cover heterogeneous configurations with a limited assumption of core heterogeneity consisting of a single big core and many smaller ones of exactly the same type. [14]. The study in [17] extended

Hill-Marty analysis to all three major speedup models. The authors of [15] evaluated the homogeneous speedup models alone. The other important issue represented by energy efficiency is demonstrated by [16] for the homogeneous and simple heterogeneous Amdahl's model.

From Table I, it can be seen that the existing models[6], [7], [8], [9], [10], [11], [12], [13], however, have a general limitation of not studying the energy-efficiency of computer system configurations, in addition to limiting any study of core heterogeneity to a simple assumption only applicable to CPU-GPU like configurations. To address these limitation, this paper makes the following *contributions*:

- extends the assumption of system core heterogeneity to cover such modern configurations as FPGA-based acceleration schemes, complex structures with many types of cores, complex Systems on Chip (SoC) including mobile computing platforms, data centers with large numbers of heterogeneous processing units, etc.;
- extends the three major speedup models (Table I) to estimate power and energy normalized speedup metrics [14], [15], [16], [17];
- studies the comparative power/performance trade-offs of these models for energy-efficient computing based on homogeneous and heterogeneous configurations;
- incorporates representations of the effects of such power and energy optimization techniques as DVFS and clock and power gating in the power models, i.e. heterogeneity in power control methods in addition to core structures;
- uses a mobile computing platform centered around ARM big.LITTLE Cortex A7-A15 cores in the form of Odroid-XU3 as a case study covering all aspects of the new modeling.

To the best of our knowledge this is the first comprehensive power and energy normalized performance analysis of the major many-core speedup models. It also represents the first attempt to extend these models to cover wider core heterogeneity. The rest of the paper is organized as follows. Section II gives the background on existing speedup models for homogeneous systems; Section III extends existing speedup models to cover a wider assumption of core heterogeneity; Section IV derives the average power consumption models for all three extended models; Section V describes a method for power and energy normalized performance analysis of these extended models for homogeneous and heterogeneous configurations; Case studies are described in the three subsequent sections; Section VI describes the experimental platform; Section VII studies the experimental platform using our models; Section VIII cross-validates these models with experiments; And Section IX concludes the paper.

II. Homogeneous Speedup Models

For a homogeneous system we consider a system consisting of N cores, each core having performance of IPS_1 instructions per second. This section describes various existing models for determining the system's speedup $SP(N)$ in relation to a single core, which can be used to find the performance of the system:

$$IPS_N = SP(N) \cdot IPS_1. \tag{1}$$

The parallel part of a workload is P and the sequential part is $(1 - P)$, This parameter reflects an application's capability of performing parallel computation. Some applications can show $P = 1$, however, in real life systems, there are always communication and shared resource access overheads that further reduce this value. Thus, this parameter is application and hardware-dependent, and is not always known. There is research on how this problem can be addressed [18]. In our models we assume that P is giver or can be determined.

A. Amdahl's Law (Fixed Workload)

The general idea of this model is to compare execution time for some fixed workload WL on a single core with the execution time for the same workload on the entire N-core system [6].

Time $T(1)$ to execute workload WL on a single core is WL/IPS_1, whereas $T(N)$ adds up the sequential execution time on one core and the parallel execution time on all N cores:

$$T(N) = \frac{(1-P)\cdot WL}{IPS_1} + \frac{P \cdot WL}{N \cdot IPS_1}, \tag{2}$$

thus the speed up can be found as follows:

$$SP(N) = \frac{T(1)}{T(N)} = \frac{1}{(1-P) + \frac{P}{N}}. \tag{3}$$

B. Gustafson's Model (Fixed Time)

Gustafson re-evaluated the fixed workload speedup model to derive a new fixed time model [7]. In this model, the workload increases with number of cores, while the execution time is fixed. An important note is that the workload scales asymmetrically: the parallel part is scaled to the number of cores, whilst the sequential part is not increased.

Let's denote the initial workload and extended workload as WL and WL' respectively. The time to execute initial workload and extended workload are $T(N)$ and $T'(N)$ respectively. The workload scaling ratio can be found from:

$$T(1) = \frac{WL}{IPS_1}, \tag{4}$$

$$T(N) = \frac{(1-P)\cdot WL}{IPS_1} + \frac{P \cdot WL'}{N \cdot IPS_1}. \tag{5}$$

and, since $T(1) = T(N)$, the extended workload can be found as:

$$WL' = N \cdot WL. \tag{6}$$

$$T'(1) = \frac{(1-P)\cdot WL}{IPS_1} + \frac{P \cdot N \cdot WL}{IPS_1}. \tag{7}$$

From the relation of scaled and unscaled execution time the following equation for speedup can be calculated:

$$SP(N) = \frac{T'(1)}{T(1)} = (1-P) + P \cdot N. \tag{8}$$

The sequential part of the workload uses one core to perform its calculation at the performance IPS_1, and the parallel execution uses N cores to perform its calculation at the performance $N \cdot IPS_1$.

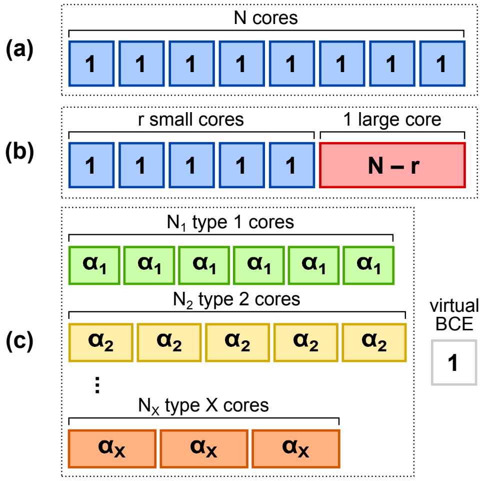

Fig. 1: The proposed extended structure of a heterogeneous system (c) compared to a homogeneous system (a) and the previous assumption [14] on heterogeneity (b). The numbers in the core boxes denote the equivalent number of BCEs.

C. Sun and Ni's Model (Memory Bounded)

Sun and Ni mixed the previous two speedup models by considering the memory bounded constraints [8], [9]. In this model the execution time and the workload change according to the memory capability. The parameter $g\left(N\right)$ reflects the scaling of the workload in relation to scaling the memory with the number of cores:

$$WL' = g\left(N\right) \cdot WL. \tag{9}$$

A typical example $g\left(N\right)$ is given for an $M \times M$ matrix multiplication, which has the memory requirement of $O(M^2)$ and the computation cost (workload) of $O(M^3)$. In this case, $g\left(N\right) = N^{\frac{3}{2}}$.

The time to execute the scaled workload can be found from (4) and (5).

$$T'\left(1\right) = \frac{\left(1-P\right) \cdot WL}{IPS_1} + \frac{P \cdot g\left(N\right) \cdot WL}{IPS_1}, \tag{10}$$

$$T'\left(N\right) = \frac{\left(1-P\right) \cdot WL}{IPS_1} + \frac{P \cdot g\left(N\right) \cdot WL}{N \cdot IPS_1}. \tag{11}$$

The speedup is calculated as follows:

$$SP\left(N\right) = \frac{T'\left(1\right)}{T'\left(N\right)} = \frac{\left(1-P\right) + P \cdot g\left(N\right)}{\left(1-P\right) + \frac{P \cdot g\left(N\right)}{N}}. \tag{12}$$

Because the workload is scaled by $g\left(N\right)$ according to (9), one of the important properties of this model is that for $g\left(N\right) = 1$ Sun and Ni's model (12) transforms into Amdahl's Law (3), and for $g\left(N\right) = N$ it becomes Gustafson's Law (8).

III. Heterogeneous Speedup Models

Previous attempts to extend speedup laws to heterogeneous systems were mainly focused on a single high performance core and many smaller cores of the same type [14], In this work we aim to to cover more diverse cases of heterogeneity pertaining to such modern architectures as ARM big.LITTLE [19] which are not directly covered by existing speedup models.

Performance-wise, the presented models describe heterogeneity using the following normal form representation. A considered heterogeneous system consists of X clusters (types) of homogeneous cores with number of cores defined as a vector $\overline{N} = (N_1, \ldots, N_X)$. Vector $\overline{\alpha} = (\alpha_1, \ldots, \alpha_X)$ defines the performance of each core by cluster (type) in relation to some base core equivalent (BCE), such that for all $1 \leq i \leq X$ we have $IPS_i = \alpha_i \cdot IPS_1$. The structure is shown in Figure 1. This section extends homogeneous speedup models for determining the heterogeneous systems speedup $SP\left(\overline{N}\right)$ in relation to a single BCE, which can then be used to find the performance of the system using (1). This representation of heterogeneity has the following limitations that have to be noted.

Drawing equivalence to a given BCE implies that there exists an equivalent representation of a workload. While the concept of the workload in homogeneous system is well defined, it is not possible to execute the same code on different heterogeneous core types unless they have the same instruction set (ISA). It may still be possible to draw equivalence, but in this paper we cover only iso-ISA systems and consider cross-ISA comparison as future development of the models.

The performance results from the real experiments, described in Section VIII, also show that the relative core performances may differ on per-instruction basis. In other words, core type i may execute instruction A faster than core type$j$, but core type j may execute instruction B faster than core type i, with both A and B in the instruction set of the BCE. Hence, on average, this will cause application- and platform-dependent $\overline{\alpha}$. Parameter P in the original homogeneous models is also application- and platform-dependent. Hence our assumption on the parameters does not reduce the applicability of the extended models in comparison to the original ones.

A. Heterogeneous Amdahl's Law (Fixed Workload)

For heterogeneous systems the combined performance of all cores while executing the parallel code P can be found as a weighted sum of all performances:

$$N_\alpha \cdot IPS_1 = \sum_{i=1}^{X} N_i \cdot \alpha_i \cdot IPS_1, \tag{13}$$

N_α is called a *performance-equivalent number* of BCEs. In other words, this performance is equal to N_α BCE cores executing the same parallel code; N_α can be a fractional number. However, in the case of synchronized-parallel execution (i.e. if the parallel execution waits for the slowest core to finish), a different equation has to be used to find N_α:

$$N_\alpha = \min \overline{\alpha} \cdot \sum_{i=1}^{X} N_i. \tag{14}$$

where $\min \overline{\alpha}$ is calculated only on the cores in use. For the model explorations in this paper (Section VII) we use (13). In the experimental validation (Section VIII) we use (14).

We also assume that the sequential part is executed on a single core in the cluster X. Hence, the time to execute the fixed workload WL on the given heterogeneous system is:

$$T_X\left(\overline{N}\right) = \frac{\left(1-P\right) \cdot WL}{\alpha_X \cdot IPS_1} + \frac{P \cdot WL}{N_\alpha \cdot IPS_1}. \tag{15}$$

The speedup in relation to single BCE is:

$$SP\left(\overline{N}\right) = \frac{T\left(1\right)}{T_X\left(\overline{N}, \overline{\alpha}\right)} = \frac{1}{\frac{(1-P)}{\alpha_X} + \frac{P}{N_\alpha}}. \tag{16}$$

B. Heterogeneous Gustafson's Model (Fixed Time)

Because of the workload scaling, we cannot directly compare speedup while executing the sequential code in the core X to single BCE execution. Let's first find the speedup $SP_X\left(\overline{N}\right)$ relative to a single core X. This is done similarly to Gustafson's derivation (Section II-B).

$$T_X\left(1\right) = \frac{WL}{\alpha_X \cdot IPS_1}, \tag{17}$$

$$T_X\left(N\right) = \frac{\left(1-P\right)WL}{\alpha_X \cdot IPS_1} + \frac{P \cdot WL'}{N_\alpha \cdot IPS_1}, \tag{18}$$

$T_X\left(1\right) = T_X\left(N\right)$, hence the extended workload can be found as:

$$WL' = \frac{N_\alpha}{\alpha_X} \cdot WL. \tag{19}$$

$$T'_X\left(1\right) = \frac{\left(1-P\right) \cdot WL}{\alpha_X \cdot IPS_1} + \frac{P \cdot N_\alpha \cdot WL}{(\alpha_X)^2 \cdot IPS_1}, \tag{20}$$

$$SP_X\left(\overline{N}\right) = \frac{T'_X\left(1\right)}{T_X\left(1\right)} = \left(1-P\right) + \frac{P \cdot N_\alpha}{\alpha_X}. \tag{21}$$

The speedup of a single core X in relation to BCE is α_X, thus the total speedup relative to BCE is:

$$SP\left(\overline{N}\right) = \alpha_X \cdot SP_X\left(\overline{N}\right) = \left(1-P\right) \cdot \alpha_X + P \cdot N_\alpha. \tag{22}$$

The sequential part of the workload uses one core in the cluster X to perform its calculation at the performance $\alpha_X \cdot IPS_1$, and the parallel execution uses all cores to perform its calculation at the performance $N_\alpha \cdot IPS_1$.

C. Heterogeneous Sun and Ni's Model (Memory Bounded)

Similarly to Amdahl's and Gustafson's cases, we can extend Sun and Ni's model as follows:

$$T'_X\left(1\right) = \frac{\left(1-P\right) \cdot WL}{\alpha_X \cdot IPS_1} + \frac{P \cdot g\left(\overline{N}\right) \cdot WL}{\alpha_X \cdot IPS_1}, \tag{23}$$

$$T'_X\left(N\right) = \frac{\left(1-P\right) \cdot WL}{\alpha_X \cdot IPS_1} + \frac{P \cdot g\left(\overline{N}\right) \cdot WL}{N_\alpha \cdot IPS_1}, \tag{24}$$

$$SP_X\left(\overline{N}\right) = \frac{T'_X\left(1\right)}{T'_X\left(N\right)} = \frac{1}{\alpha_X} \cdot \frac{\left(1-P\right) + P \cdot g\left(\overline{N}\right)}{\frac{(1-P)}{\alpha_X} + \frac{P \cdot g\left(\overline{N}\right)}{N_\alpha}}. \tag{25}$$

The speedup in a heterogeneous system relative to BCE is calculated as follows:

$$SP\left(\overline{N}\right) = \alpha_X \cdot SP_X\left(\overline{N}\right) = \frac{\left(1-P\right) + P \cdot g\left(\overline{N}\right)}{\frac{(1-P)}{\alpha_X} + \frac{P \cdot g\left(\overline{N}\right)}{N_\alpha}}. \tag{26}$$

When $g\left(\overline{N}\right) = 1$, this model transforms into heterogeneous Amdahl's Law (16), and for $g\left(\overline{N}\right) = \frac{N_\alpha}{\alpha_X}$ it becomes heterogeneous Gustafson's Law (22), as expected from (19).

For all heterogeneous models, substitution $\alpha_X = 1, N_\alpha = N$ will give the homogeneous versions of respective models. In other words, homogeneity is a special case of heterogeneity.

IV. Average Power Consumption Models

The power consumption models are built under the assumption that the cores consume power when idle. When idle power is zero, this assumption covers the special case of power gating.

Let's the active power of a core in the homogeneous system (Section II) be W_A and the idle power of a core be W_i respectively. Active power can also be expressed as a sum of idle power and effective power W_1 (used for computation), $W_A = W_1 + W_i$. In the total power consumption of the system, the constant term of total idle power W_{idle} does not benefit to the model and can be added later. The power models $W\left(N\right)$ are focused on the effective power, and the total power of the system can be calculated as follows:

$$W_{total} = W\left(N\right) + W_{idle}, \tag{27}$$

In the normal from representation of a heterogeneous system (Section III), the difference between power consumptions of the cores is expressed by the vector $\overline{\beta} = (\beta_1, \ldots, \beta_X)$, which defines the effective power in relation to a BCE's effective power, such that for all $1 \leq j \leq X$ we have effective power $W_j = \beta_j \cdot W_1$. All idle powers of heterogeneous cores are combined into W_{idle}. In the general case, we say that:

$$W_{idle} = N_i \cdot W_i, \tag{28}$$

where N_i is idle power equivalent number of BCEs and W_i is the idle power of a single BCE.

The effective power model can be found as a time-weighted average of the sequential power W_S the and parallel power W_P:

$$W\left(N\right) = \frac{W_S \cdot T_S\left(N\right) + W_P \cdot T_P\left(N\right)}{T_S\left(N\right) + T_P\left(N\right)}, \tag{29}$$

where $T_S\left(N\right)$ and $T_P\left(N\right)$ are speedup-dependent times to execute sequential and parallel parts respectively.

In the homogeneous system:

$$W_S = W_1, \; W_P = N \cdot W_1. \tag{30}$$

In the heterogeneous system, if we execute the sequential code on a single core X:

$$\begin{aligned} W_S &= \beta_X \cdot W_1, \\ W_P &= W_1 \cdot \textstyle\sum_{j=1}^{X} \beta_j \cdot N_j = N_\beta \cdot W_1, \end{aligned} \tag{31}$$

for average case of parallel execution (13) . For synchronized-parallel execution (14), N_β is calculated as follows:

$$N_\beta = \min \overline{\alpha} \cdot \sum_{i=1}^{X} \frac{N_i \beta_i}{\alpha_i}. \tag{32}$$

N_β is called a *power-equivalent number* of BCEs. Heterogeneous power models will transform into homogeneous if $\alpha_X = \beta_X = 1$ and $N_\alpha = N_\beta = N$.

A. Power Model for Amdahl's Law (Fixed Workload)

From (15) we know that:

$$T_S\left(\overline{N}\right) = \frac{\left(1-P\right) \cdot WL}{\alpha_X \cdot IPS_1}, \; T_P\left(\overline{N}\right) = \frac{P \cdot WL}{N_\alpha \cdot IPS_1}. \tag{33}$$

By substituting (33) and (31) into (29) we have a power model for the heterogeneous system:

$$W\left(\overline{N}\right) = \left(\frac{\beta_X}{\alpha_X} \cdot \left(1-P\right) + \frac{N_\beta}{N_\alpha} \cdot P\right) \cdot SP\left(\overline{N}\right) \cdot W_1, \tag{34}$$

where the speedup $SP\left(\overline{N}\right)$ is calculated using (16). For homogeneous system, this will transform into:

$$W\left(N\right)=SP\left(N\right)\cdot W_1, \tag{35}$$

thus for Amdahl's Law the power scales with the speedup.

B. Power Model for Gustafson's Model (Fixed Time)

In this model we have a fixed time T, so the workload splits execution into:

$$T_S\left(\overline{N}\right)=(1-P)\cdot T,\ T_P\left(\overline{N}\right)=P\cdot T. \tag{36}$$

Thus, we can find a power model for the heterogeneous system:

$$W\left(\overline{N}\right)=\left(\frac{\beta_X\cdot(1-P)+N_\beta\cdot P}{\alpha_X\cdot(1-P)+N_\alpha\cdot P}\right)\cdot SP\left(\overline{N}\right)\cdot W_1, \tag{37}$$

For homogeneous system, this will transform into:

$$W\left(N\right)=SP\left(N\right)\cdot W_1, \tag{38}$$

where the speedup $SP\left(\overline{N}\right)$ is calculated using (22).

C. Power Model for Sun and Ni's Model (Memory Bounded)

From (24) we can find:

$$T_S\left(\overline{N}\right)=\frac{(1-P)\cdot WL}{\alpha_X\cdot IPS_1},\ T_P\left(\overline{N}\right)=\frac{P\cdot g\left(\overline{N}\right)\cdot WL}{N_\alpha\cdot IPS_1}. \tag{39}$$

Thus, we can find a power model for the heterogeneous system:

$$W\left(\overline{N}\right)=\left(\frac{\frac{\beta_X}{\alpha_X}\cdot(1-P)+\frac{N_\beta}{N_\alpha}\cdot P\cdot g\left(\overline{N}\right)}{(1-P)+P\cdot g\left(\overline{N}\right)}\right)\cdot SP\left(\overline{N}\right)\cdot W_1, \tag{40}$$

where the speedup $SP\left(\overline{N}\right)$ is calculated using (26). This model will transform into (34) if $g\left(\overline{N}\right)=1$, or (37) for $g\left(\overline{N}\right)=\frac{N_\alpha}{\alpha_X}$. For homogeneous system, (40) will also transform into:

$$W\left(N\right)=SP\left(N\right)\cdot W_1. \tag{41}$$

All power models – (34), (37), and (40) – can be represented using *power scaling* $PS\left(\overline{N}\right)$, which can be derived from the respective model equations:

$$W\left(\overline{N}\right)=PS\left(\overline{N}\right)\cdot SP\left(\overline{N}\right)\cdot W_1. \tag{42}$$

V. Power-normalized and Energy-normalized Performance

The power model explains the total power consumption in this model during workload execution. It is likewise represents the cooling capacity. Furthermore, it is simple to model the performance achievable at the same cooling capacity from calculating performance per watt (Perf/Watt). This model is reciprocal of energy per instruction (EPI_N) because performance is the reciprocal of execution time.

EPI_N can be found from dividing the total power (27) by the system's performance (1):

$$EPI_N=\frac{W_{total}}{IPS_N}=\frac{W\left(\overline{N}\right)+W_{idle}}{IPS_1\cdot SP\left(\overline{N}\right)}, \tag{43}$$

which is true for all cases of $W\left(\overline{N}\right)$: Amdahl's, Gustafson's, or Sun and Ni's.

For a single BCE we can denote energy per instruction as a sum of effective energy EPS_1 and idle energy EPS_i:

$$EPI_{BCE}=\frac{W_1}{IPS_1}+\frac{W_i}{IPS_1}. \tag{44}$$

Applying the power model (42) to (43) and also considering (28), we find:

$$EPI_N=EPI_1\cdot PS\left(\overline{N}\right)+\frac{N_i\cdot EPI_i}{SP\left(\overline{N}\right)}. \tag{45}$$

This equation shows that the effective component of the energy increases with the power scaling $PS\left(\overline{N}\right)$, and the idle energy decreases with the speedup $SP\left(\overline{N}\right)$.

Energy-normalized performance represents how much performance one can gain if willing to increase energy per operation. This gain in relation to BCE can be found as:

$$\left(\frac{IPS_N}{EPI_N}\right)\cdot\left(\frac{IPS_1}{EPI_{BCE}}\right)^{-1}=SP\left(\overline{N}\right)\cdot\frac{EPI_N}{EPI_{BCE}}. \tag{46}$$

The equation shows that the increment factor scales with the speedup, and this is true for all three models.

VI. Experimental Platform

We carry out an extensive case study demonstrating the use of these models. This study is based on a multi-core mobile platform, the Odroid-XU3 board [19]. The main part of it is the 28nm Application Processor Exynos 5422. It is an SoC hosting an ARM big.LITTLE heterogeneous octa-core processor consisting of four Cortex A7 cores and four Cortex A15 cores. The big Cortex-A15 is a high performance 32-bit core having 32 KB instruction and 32KB data L1 caches and 2 MB L2 cache and the maximum frequency of 2.0 GHz. The LITTLE Cortex-A7 is a low power 32-bit core including the same L1 cache size and 512 KB L2 cache, and the maximum frequency of 1.4 GHz.

This SoC also has four power domains: A7 power domain, A15 power domain, GPU and memory power domains. The Odroid-XU3 board allows per-domain DVFS using voltage-frequency pairs, however for frequencies within the range of 200MHz to 800MHz, the voltage stays constant (DFS-only).

The traditional simple assumption for heterogeneous architectures, shown in ure 1(b), cannot describe systems such as big.LITTLE. Our models do not suffer from the same restrictions and can be applied to big.LITTLE and similar structures.

VII. Model Exploration

Models of the Odroid-XU3 platform have been created using our methods and these models are used to calculate predicted system behaviors under various operating conditions. Metrics including speedup, performance, power, energy per operation, etc. are obtained from exploring with models.

A. Calculating Parameters

A set of characterization experiments was carried out to determine power consumptions and performances for each core type. The main parameters for this study can be arranged into the following points.

System's heterogeneity: Following the extended heterogeneous system structure assumption proposed in Section III, we can set the constants for Odroid-XU3. Two types of cores (A7 and A15) give us $X = 2$. In our experiments, in order to improve measurement accuracy, one of the A7 cores was reserved for exclusive use by the operating system. Therefore the numbers of cores by type are $N_{A7} = 3$ and $N_{A15} = 4$.

Core active powers: Theoretical active power calculations are derived from the experiments. The power is measured while executing a full workload on the processor's cores and sweeping through all DVFS points [20]. In general, the theoretical dynamic power estimation can be calculated by the power equation [21]:

$$Power_{Dynamic} = C \cdot V^2 \cdot F, \qquad (47)$$

where V is the voltage, F is the frequency, and C is the constant, which relates to the combined capacitance of the switching logic. We use the experimental data to curve-fit by MATLAB and derive the Cortex A7 and A15 power equations. The result is the following values for C: in A7 it is equal to 0.127nF and in A15 it is 0.599nF, with R-squared values greater than 0.99. Considering A7 as BCE, we have $\beta_{A7} = 1, \beta_{A15} = 4.71$ to supply our power models from SectionIV.

Core idle powers: Theoretical idle power calculations supported by the experiments [20] give the value of A15 idle power as $W_{iA15} = 0.021W$. Having one of the A7 cores occupied by the operating system prevents measuring idle power for that domain. In this study we accept the minimum measured power of 0.008W as the domain's idle power W_{iA7}, which is also our BCE's idle power W_i. We also do not switch off the cores, so the total idle power of the system remains constant: $W_{idle} = N_i \cdot W_i$. From the above measurements, we calculate $N_i = 3.625$.

Relative performance of cores: We did not use performance counters to find the actual number of clocks per instruction (CPI) for different types of instructions. Given these are RISC processors, we assume the general value of $CPI = 1$ without losing generality.

From the characterization experiments, we found that on the average an A15 core has 1.5 times the throughput of an A7 core when both are running at the same frequency. We also want to calculate the models for different DVFS points. However, for frequency values when A7 cannot be run and A15 can, or for the DFS-only region, the performance specifying vector $\overline{\alpha}$ changes. Therefore, for each DVFS point the value for α_{A15} is computed as follows:

$$\alpha_{A15} = 1.5 \cdot \frac{freq_{A15}}{freq_{A7}}. \qquad (48)$$

We assume $\alpha_{A7} = 1$ considering A7 is our BCE.

Parallelization parameter: The speedup models take parameter P from the nature of the executed workload. In our theoretical calculations we investigate a number of values for P. In the next section we provide results for two example values ($P = 0.9$ and $P = 0.1$) covering highly parallelizable and not parallelizable cases.

B. Outcomes

In this section we present selected calculation results organized in three groups. More comprehensive set of results can be found in [22].

Metric explorations on a fixed DVFS point: In the first group of results we present the following metrics of interest calculated for a various combination of active and idle cores on a fixed DVFS point ($freq_{A7} = freq_{A15} = 1400$MHz):

- Performance IPS_N according to (1),
- Average power consumption W_{total} according to (27),
- Energy per instruction EPI_N according to (45),
- Energy-normalized performance according to (46).

These parameters have been estimated for all presented heterogeneous speedup and power models. The graphs for different models display similar trends, hence we only present Sun and Ni's model for its generality; $g(N)$ was set to the matrix multiplication example presented in Section II:

$$g\left(\overline{N}\right) = \left(\frac{N_\alpha}{\alpha_{A15}}\right)^{\frac{3}{2}}.$$

Figure 2 shows the graphs for the listed parameters for $P = 0.9$ and $P = 0.1$.

It can be seen from the data that although the power consumption increases with the number of cores participating in the computation, the performance also increases with the cumulative effect being that the performance per unit of power spent still improving with more cores used. This is mainly because of the influence of idle power. If you don't use a core, the idle power is wasted.

The higher the parallelization factor, the better the performance and energy-normalized performance, as expected.

More interestingly, from the energy per instruction metric one can see it increasing when the number of A15 cores increase but decrease when the number of A7 cores increase. This is on account of the much higher efficiency of A7 cores in terms of energy per instruction.

Homogeneous example: This group of results illustrate the scaling of the energy per instruction EPI_N and the energy-normalized performance with the system's frequency. The values for frequencies have been selected within the allowed range of 200MHz to 2000MHz and the same frequency have been set for A7 and A15 cores if possible (for values above 1400MHz, the frequency for A7 is set to the allowed maximum of 1400MHz). This point causes a non-smooth change in α_{A15} leading to a peculiar non-smooth behavior of the metrics. There are two other less obvious behavior boundaries: 800MHz, where DVFS switches to DFS, and 1900MHz, above which the A15 cores experience throttling because of thermal issues. All these points are reflected by our models.

Figure 3 shows the graphs for the energy per instruction EPI_N in different heterogeneous core combinations for $P = 0.9$. Figure 4 shows the graphs for the energy-normalized performance, also for $P = 0.9$.

Homogeneous example: Figure 5 presents an example of applying the presented models to a homogeneous system for completeness, demonstrating that $X = 1$ also works. From this figure, we can make an interesting observation: if you put more cores to solving a problem with a low parallelization capability

($P = 0.1$), energy per instruction suffers, especially at the lower frequencies.

VIII. Cross-Validation

The models operate on application- and platform-dependent parameters, which are typically unknown and imply high effort in characterization. However, in order to prove that the proposed models work, it is sufficient to show that, if $\overline{\alpha}$, $\overline{\beta}$ and P are defined, the performance and power behavior of the system follows the model's prediction. These parameters can be fixed by a set of synthetic benchmarks.

The proper validation of the applicability of the proposed modeling will require extensive runs of standard benchmark suites and real-life applications, and is a subject of future work. Also, in this paper we present cross-validation results only for the Amdahl's model extension.

A. Calculating parameters

The model characterization is derived from single core experiments. These characterized models are used to predict multi-core execution in different core configurations. These predictions are then cross-validated against experimental results.

In our experiments we use three benchmarks: square root calculation (sqrt), integer arithmetic (int), and logarithm calculation (log). The characterization and cross-validation is done separately for each benchmark. Hence we derive three sets of parameters.

Parallelization parameter: The benchmarks have been developed specifically for this experiment in order to provide control over the parallelization parameter P. Hence, P is not a measured parameter, but a control parameter that tells the application the ratio of distribution between the parallel (multi-threaded) and sequential (single thread) execution.

The benchmarks implement synchronized-parallel execution, hence the models for these benchmarks should use N_a and N_b calculated according to (14) and (32) respectively.

Relative performance of cores: All experiments in this section are run with both A7 and A15 cores at 1400MHz. In this study, we set BCE to A7, hence $\alpha_{A7} = 1$; and α_{A15} can be found as a ratio of execution times $\alpha_{A15} = T_{A7}/T_{A15}$, as shown in Table II. It can be seen that, A15 is expectedly faster than A7 for integer arithmetic and logarithm calculation, however square root calculation is faster on A7. This is confirmed multiple times in many experiments [20]. Three different benchmarks provide different α_{A15} values, which strengthens our study.

Core idle and active powers: Idle powers are determined as average over 1min of measurements while the platform is running only the operating system. The idle power values are the same as in Section VII : $W_{iA7} = 0.1363W$ and $W_{iA15} = 0.3759W$, which are used for all benchmarks. The standard deviation during the idle power measurements is 1.949% of the mean value.

Effective powers W_{A7}, W_{A15} are calculated from measured active powers by subtracting idle power according to (27). The power ratios are then found as $\beta_{A7} = 1$ and $\beta_{A15} = W_{A15}/W_{A7}$; the values are presented in Table II.

B. Outcomes

A large number of experiments have been carried out covering all benchmarks (sqrt, int, and log) and all core configurations, and repeated for $P = 0.3$ and $P = 0.9$.

Model predictions and experimental measurements for selected homogeneous and heterogeneous multi-core configurations can be found in Table III. The measured speedup is calculated as time for a single A7 core execution (Table II) over the benchmark's measured execution time.

An important observation is that the differences between the model predictions and experimental measurements are very small. For the results that are not presented, the error values are similar. The speedup error never exceeds 1%, and the power error never exceeds 4%, which is comparable to the standard deviation of the characterization measurements.

A possible explanation for the low error values can be that the used synthetic benchmarks produce constant $\overline{\alpha}$, $\overline{\beta}$ and P values: P is constant as it is a control parameter; $\overline{\alpha}$, $\overline{\beta}$ are constants because the benchmarks are based on repeating the same calculation. In real life applications, these should have at least some degree of variability. However, these small errors also prove that the models can be used with high confidence if it is possible to track these parameters.

A counter intuitive result for 7-core (three A7 cores and four A15 cores) execution having lower power consumption than four A15 cores and no A7 cores can be explained by synchronized-parallel execution. Because the parallel workload is equally split between these cores, the A15 cores finish early and wait for A7 cores. This idling reduces the average total power consumption, however it implies that intelligent workload distribution can improve core utilization by scheduling more tasks to A15 cores than to A7 ones so that they finish at the same time. This is a possible objective for future research.

IX. Conclusions

This paper addresses the emerging issue of the system heterogeneity becoming more common and diverse in its structure well beyond the traditional CPU+GPU assumption. This is done by introducing the extended model for system heterogeneity. The three known speedup models (Amdahl's Law, Gustafson's model, Sun and Ni's model) are extended to cover this wider heterogeneity. In addition to performance speedup, the paper presents the models for power and energy related system metrics.

The derived theoretical models have been applied to a real-life heterogeneous system, whose structure does not fit into the traditional heterogeneity assumption. The model parameters have been characterized from a set of experiments, and the metrics of interest have been calculated to demonstrate the model capabilities. These metrics include speedup, average power scaling, energy per instruction and energy-normalized performance. Cross-validations comparing model results to experimental results show very small errors, typically below 1% for speedup and below 4% for power.

Acknowledgment

This work is part of the PRiME project (EPSRC grant EP/K034448/1). The authors thank Rem Gensh and Ali Aalsaud for providing some of the Odroid-XU3 experimental data. The

TABLE II: Characterization experiments: single core execution

benchmark	sqrt		int		log	
fixed workload, iterations	$2.4 \cdot 10^8$		$4.08 \cdot 10^9$		$2.4 \cdot 10^8$	
core type i	A7	A15	A7	A15	A7	A15
measured time, ms	75020	79892	79329	64046	62927	35711
measured active power, W	0.2563	0.8407	0.2620	0.8418	0.2874	0.9406
power measurement std dev	3.934%	3.617%	4.919%	4.392%	4.519%	6.530%
calculated effective power, W	0.1200	0.4648	0.1257	0.4659	0.1511	0.5647
α_i	1	0.9390	1	1.2386	1	1.7621
β_i	1	3.8733	1	3.7064	1	3.7373

TABLE III: Initial cross-validation results for heterogeneous Amdahl's Law

				time, ms	speedup			average total power, W		
bench	P	N_{A7}	N_{A15}	measured	predicted	measured	error	predicted	measured	error
sqrt	0.3	3	0	59992	1.2500	1.2505	0.04%	0.6622	0.6686	0.96%
sqrt	0.3	0	4	61911	1.2116	1.2117	0.01%	1.1119	1.0993	1.15%
sqrt	0.3	2	2	61910	1.2116	1.2118	0.01%	1.0438	1.0312	1.22%
sqrt	0.3	3	4	59359	1.2641	1.2638	0.02%	1.0769	1.0666	0.97%
sqrt	0.9	3	0	29988	2.5000	2.5017	0.07%	0.8122	0.8042	0.99%
sqrt	0.9	0	4	25977	2.8893	2.8879	0.05%	1.9424	1.9252	0.89%
sqrt	0.9	2	2	25961	2.8893	2.8897	0.02%	1.4548	1.4239	2.17%
sqrt	0.9	3	4	18300	4.1082	4.0995	0.21%	1.9515	1.9403	0.58%
int	0.9	3	0	31705	2.5000	2.5021	0.08%	0.8265	0.8305	0.49%
int	0.9	0	4	20823	3.8112	3.8097	0.04%	1.9457	1.9351	0.55%
int	0.9	2	2	24264	3.2708	3.2694	0.04%	1.3739	1.3537	1.49%
int	0.9	3	4	16637	4.7777	4.7682	0.20%	1.8478	1.8117	1.99%
log	0.9	3	0	25118	2.5000	2.5053	0.21%	0.8900	0.8925	0.29%
log	0.9	0	4	11580	5.4219	5.4341	0.22%	2.2497	2.2135	1.64%
log	0.9	2	2	17722	3.5492	3.5508	0.04%	1.3791	1.3615	1.29%
log	0.9	3	4	11690	5.3960	5.3830	0.24%	1.8889	1.8570	1.72%

first author thanks his sponsor HCED-Iraq and the University of Technology - Iraq for funding and other support.

References

[1] S. Borkar, "Thousand core chips: a technology perspective," in *Proceedings of the 44th annual Design Automation Conference*. ACM, 2007, pp. 746–749.

[2] R. H. Dennard, V. Rideout, E. Bassous, and A. Leblanc, "Design of ion-implanted mosfet's with very small physical dimensions," *Solid-State Circuits, IEEE Journal of*, vol. 9, no. 5, pp. 256–268, 1974.

[3] G. E. Moore *et al.*, "Cramming more components onto integrated circuits," *Proceedings of the IEEE*, vol. 86, no. 1, pp. 82–85, 1998.

[4] J. G. Koomey, S. Berard, M. Sanchez, and H. Wong, "Implications of historical trends in the electrical efficiency of computing," *Annals of the History of Computing, IEEE*, vol. 33, no. 3, pp. 46–54, 2011.

[5] F. J. Pollack, "New microarchitecture challenges in the coming generations of cmos process technologies (keynote address)," in *Proceedings of the 32nd annual ACM/IEEE international symposium on Microarchitecture*. IEEE Computer Society, 1999, p. 2.

[6] G. M. Amdahl, "Validity of the single processor approach to achieving large scale computing capabilities," in *Proceedings of the April 18-20, 1967, spring joint computer conference*. ACM, 1967, pp. 483–485.

[7] J. L. Gustafson, "Reevaluating amdahl's law," *Communications of the ACM*, vol. 31, no. 5, pp. 532–533, 1988.

[8] X.-H. Sun and L. M. Ni, "Another view on parallel speedup," in *Supercomputing'90., Proceedings of*. IEEE, 1990, pp. 324–333.

[9] ——, "Scalable problems and memory-bounded speedup," *Journal of Parallel and Distributed Computing*, vol. 19, no. 1, pp. 27–37, 1993.

[10] J. W. Tschanz, S. G. Narendra, Y. Ye, B. Bloechel, S. Borkar, V. De *et al.*, "Dynamic sleep transistor and body bias for active leakage power control of microprocessors," *Solid-State Circuits, IEEE Journal of*, vol. 38, no. 11, pp. 1838–1845, 2003.

[11] S. Eyerman and L. Eeckhout, "Fine-grained dvfs using on-chip regulators," *ACM Transactions on Architecture and Code Optimization (TACO)*, vol. 8, no. 1, p. 1, 2011.

[12] A. Das, M. Schuchhardt, N. Hardavellas, G. Memik, and A. Choudhary, "Dynamic directories: A mechanism for reducing on-chip interconnect power in multicores," in *Proceedings of the Conference on Design, Automation and Test in Europe*. EDA Consortium, 2012, pp. 479–484.

[13] T. S. Muthukaruppan, A. Pathania, and T. Mitra, "Price theory based power management for heterogeneous multi-cores." in *ASPLOS*, 2014, pp. 161–176.

[14] M. D. Hill and M. R. Marty, "Amdahl's law in the multicore era," *Computer*, no. 7, pp. 33–38, 2008.

[15] N. Ye, Z. Hao, and X. Xie, "The speedup model for manycore processor," in *Information Science and Cloud Computing Companion (ISCC-C), 2013 International Conference on*. IEEE, 2013, pp. 469–474.

[16] D. H. Woo and H.-H. S. Lee, "Extending amdahl's law for energy-efficient computing in the many-core era," *Computer*, no. 12, pp. 24–31, 2008.

[17] X.-H. Sun and Y. Chen, "Reevaluating amdahls law in the multicore era," *Journal of Parallel and Distributed Computing*, vol. 70, no. 2, pp. 183–188, 2010.

[18] A. B. Downey, "A model for speedup of parallel programs," EECS Department, University of California, Berkeley, Tech. Rep. UCB/CSD-97-933, Jan 1997. [Online]. Available: http://www.eecs.berkeley.edu/Pubs/TechRpts/1997/5394.html

[19] (2015) Odroid platform. [Online]. Available: http://www.hardkernel.com/main/products/

[20] R. Gensh *et al.*, "Experiments with odroid-xu3 board," Newcastle University, Computing Science, Claremont Tower, Claremont Road,Newcastle Upon Tyne, NE1 7RU, England., Tech. Rep., 2015.

[21] J. Rabaey and M. Pedram, *Low Power Design Methodologies*, ser. The Springer International Series in Engineering and Computer Science. Springer US, 2012. [Online]. Available: https://books.google.co.uk/books?id=9IzuBwAAQBAJ

[22] M. A. N. Al-hayanni *et al.*, "Extended speedup models into power and energy normalized performance models for energy-efficient many-core processors," Newcastle University, Tech. Rep. NCL-EEE-MICRO-TR-2016-198, 2016.

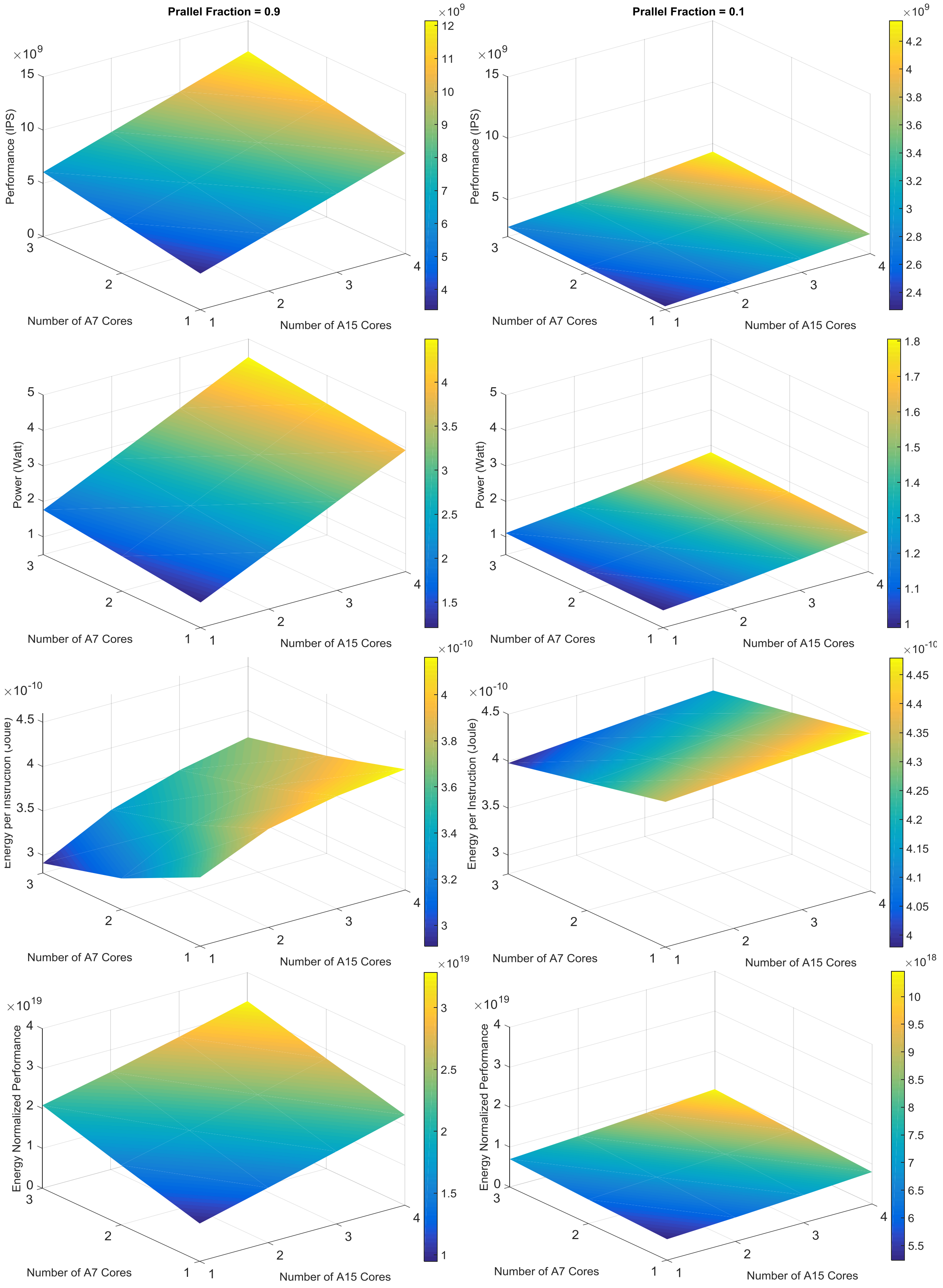

Fig. 2: Model exploration on a single DVFS point (1400MHz), two different parallelization ratios P=0.1 and 0.9, and core configurations with 1-3 A7 cores and 1-4 A15 cores. The metrics compared include performance, power, energy per instruction and energy normalized performance.

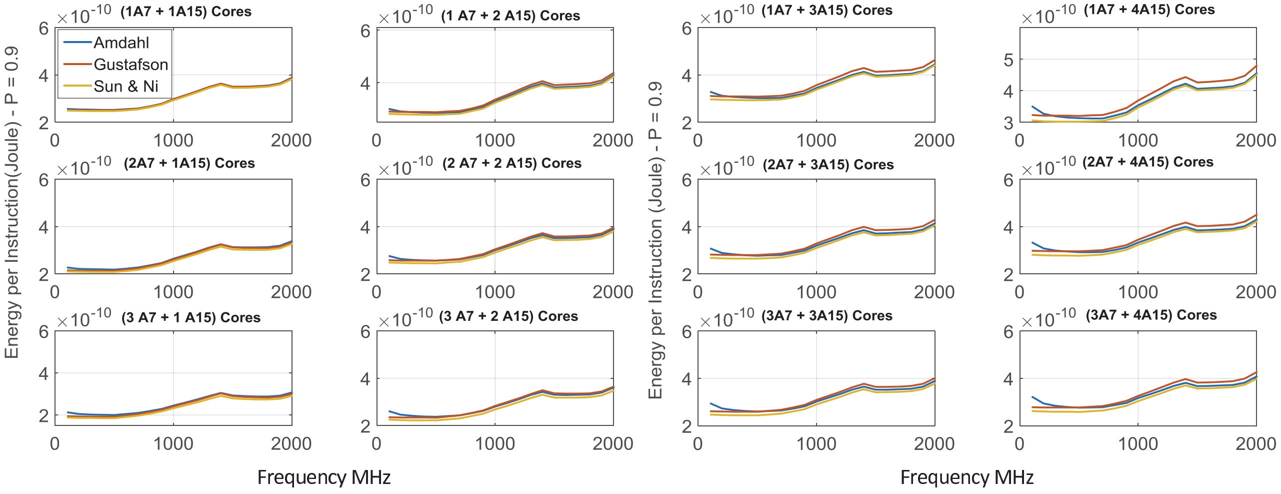

Fig. 3: Energy per instruction, exploring through a number of core combinations and frequencies from 200MHz to 2GHz, with P=0.9.

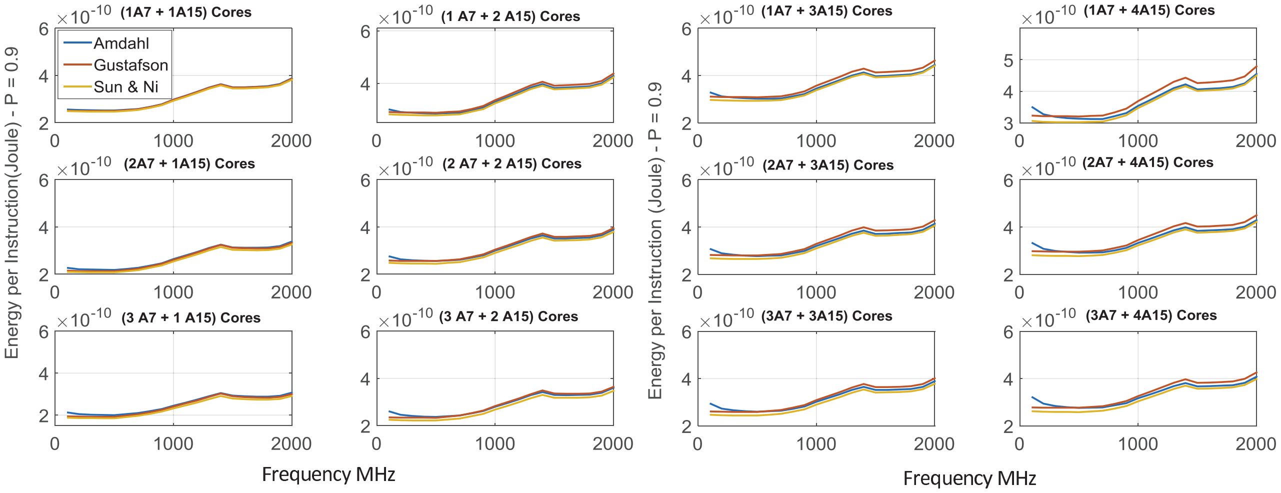

Fig. 4: Energy normalized performance, exploring through a number of core combinations and frequencies from 200MHz to 2GHz, with P=0.9.

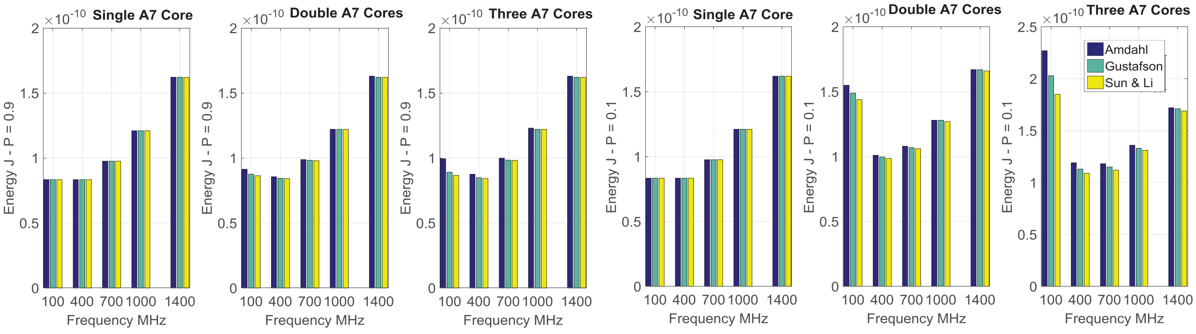

Fig. 5: Energy per operation, comparing the three different models and P=0.1 and P=0.9. Homogeneous systems with between one and three A7 cores and zero A15 cores running.